Radical Technologies
Project Based IT Training & Certification Centre

MASTER'S PROGRAM IN FULL STACK JAVA

> “SUCCESS IS, IN FRONT OF YOUR EYES!”

Integrated with International PG Diploma From UK Government Authorised Awarding Body. This masterprogram is Endorsed by OTHM and You will achieve the International Post Graduate Diploma Certificate in Software Development & Automation Engineering

WHAT WE COVER FROM BASIC TO ADVANCED

Complete In-depth Knowledge

- Core Java
- Advanced Java
- Advanced Framework with Microservices
- MEAN Stack
- Html | CCS5 | Java Script
- AWS Cloud Development

JOB ORIENTED SCENARIOS

Multiple Job Oriented Responsibilities Practically Explained –Gain Knowledge of 5+ Year Experienced Full Stack Developer. 400+Hrs of content.

YES YOU CAN BE ONE OF THESE!

Full Stack Developer

Mean Stack Developer

Software Engineer

Software Developer

Front End Developer

Backend Developer

Our students placed in various Pay packages in MNC's Ranging from **5 Lakhs to 25 Lakhs**. Experience Level from **freshers to 18+** Year experienced candidates

DURATION

6 Months Learning

+ 6 Months Projects, Assessments and Assignments

MAJOR FEATURES OF THE COURSE

Major achievements of this program is that, This program is integrated with International PG Diploma Certificate Program in Software Development & Automation Engineering –Awarded by OTHM – UK, Ofqual Approved UK Qualification provider (https://www.gov.uk/government/organisations/ofqual) Program Include Major and Minor Projects with all portions practically deeply explained, 200+ Assignments and 50 hours of Mock interview session. You will acquire a knowledge of 5+ year experienced in Full Stack Development after this course This program is good for freshers and up to 3 – 4 years of industry experienced professional & Those who have career gap

CERTIFICATES ACHIEVED

You will receive 4 certificates after successful completion.

1. RCITP after 6 months
2. Internship Certificate explaining Projects undergone
3. International PG Diploma certificate in Software Development & Automation Engineering - OTHM UK After 1 year of successful completion of the course. You have to undergo multiple assignments to achieve the International PG Diploma. After successful completion of assignment, you have to appear for 12

assessments, on topics covered during the course . After fulfilling all the above criteria, International PG Diploma certificate in Software Development & Automation Engineering is awarded directly by OTHM

4. Fill any Career gap with our consulting companies who provides documentation on the projects completed.

SALARY :- Avg. salary of a Full stack , Software Developer range from 2.8 Lakh and goes up to 15 Lakhs per annum for 5+ year experienced developer.

PROGRAM HIGHLIGHT

- 18000+ Students Empowered till now
- Online/Classroom/Self-Paced
- 6 months Intensive Training followed by 6 Month Assessments and Assignments
- Program include Four Certification and Career Gap Documentation.
- Start Date – Every Month One Batch (Per Batch maximum 10 Students)
- 450+ Hiring Partners
- Certified from OTHM–UK Recognized by ofqual.gov.uk
- Minimum 5 Interview Call and until you get Job
- What you get After the Training – Gain Knowledge of At least 5+ Year Experienced Professional
- Designed for Freshers to Working Professionals
- 360 Degree Career Support
- One-on-One with Industry Mentors
- Dedicated Student Mentor
- 0 % Interest EMI – No Interest on EMI Paying – Approximately 6000 per month.
- Project Driven industry mentorship in Full Stack Application Development in Java

- Dedicated career support,
- Learn 30 + Application tools.
- 300+ core project &amp; assignment activities
- 200 + Assignments
- 50 hours of Mock interview session
- Dedicated Trainers with ample of Industry Experience
- Project based IT Training and Certification programs.
- You Gain Knowledge of 5+ Full Stack Java Developer
- Industry standard Projects & Assignments with Mentor Support
- Lab Access and Practical Guidance

TARGET AUDIENCE?

Those who would like to go with programming technologies, Coding etc. Engineers, Freshers, Domain Experts, Software & IT Professionals, Those who having career gap & Interested in Programming Technologies and Coding etc. Any One from Non-technical and Technical Background who wants to come into IT Engineering/Management Graduate or Post-graduate Fresher Students.

MINIMUM ELIGIBILITY

Any Bachelor’s degree. Completed or not completed . No coding experience required . If you have any Educational Gap or any other career gap , you can do this program to boost up your career. The qualifications provided by UK Regulatory board is equal to Level 7 Masters Degree in UK.

HOW YOU BENEFIT FROM THIS PROGRAM

International Post Graduate Diploma Certificate without quitting your job Cutting-edge curriculum designed by industry experts. Trainers are from Real Time Industry Career transition with 30 to 100% salary hike

FREQUENTLY ASKED QUESTIONS

1.Who Should do this course

Freshers, Any Graduate, 2 to 4 Years Experienced up skilling enthusiasts. 3rd Year Graduates who are going to attend campus Interviews. These courses designed in a way to be suitable for all branches of Engineering and all type of graduates – Science and non-science Graduates. Option to customise the subject according to the interest of candidate is also available

2.How many Interview Calls You will provide

Minimum 5 Interview Calls and Maximum , until you get the JOB

3.Do I get Refund, if I am not interested to continued?

Yes. you should inform the Administrator and send email to **masters@radicaltechnologies.co.in** to cancel your admission within 7 days of the enrolment. Once we allocate the trainer and further proceedings, we will not be able to refund.

4.How I will get Industry knowledge?

All our programs are developed by Industry person, who is having 10+ years real working knowledge. Under their mentorship, you will be able to gain at least 5+ year experience

5.Is this is 100% Job Guarantee program ? And how is the Salary Structure?

Yes, it's a 100% JOB Guaranteed program. Minimum 5 Interview calls and upto you get satisfied Job. Minimum salary Package of 5 Lakhs per annum for the right candidate.

6.Is there any limitation on number of Interview calls providing?

No, You will get calls until you get placed. And we will provide unlimited interview calls after you get placed also.

7.How many people got placed as part of this program?

We have placement record of over 6000+, Out of this 800+ are from Master Program.

CORE JAVA

Introduction To Java – 2 hr

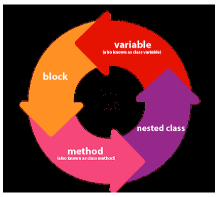
- History of Java
- Java Flavors,
- What is Java ,
- characteristics
- JVM Architecture
- Bytecode
- Class Loader
- Unicode
- Classpath
- Path

Fundamentals of Java Programming – 4 hrs

- Obect oriented concepts
- Keywords, Datatypes, Variables, Operators, Casting
- Selection statement (if, switch)
- Control statements (while, do while , for)
- Conditional statements
- static
- Arrays

Object Oriented Programming with Java – 5 hrs

- Classes and Objects
- Structure of a class – its internals (Data Members, methods)
- Using static
- Constructor
- this keyword
- modifiers
- playing with the object (copying, casting)
- Garbage collection
- Abstract class



Inheritance – 3 hrs

- Basics (extends keyword)
- Modifiers and their scope
- Deriving a class
- super, final keyword
- why java does not support multiple inheritance?

Polymorphism – 2hrs

- overloading a method
- overloading a constructor
- method overriding
- accessing base class method

Packages and Interfaces – 2hrs

- basics
- modifiers and their scope chart
- setting classpath
- compiling and accessing a packaged class
- types of packages

- user defined package

Exploring java.lang package – 2hrs

- String, StringBuffer, Arrays,
- Wrapper classes

Exception Handling – 3 hrs

- Basics
- Hierarchy of exceptions
- Handling exception – Try, catch, finally, throw, throws
- User defined exceptions

Threads – 3 hrs

- Basics, Thread class , Runnable Interface
- Thread model
- Life cycle – start(), run()
- Scheduling
- Deadlocks / concurrency issues
- Synchronization – as a block, as a modifier
- Daemon thread

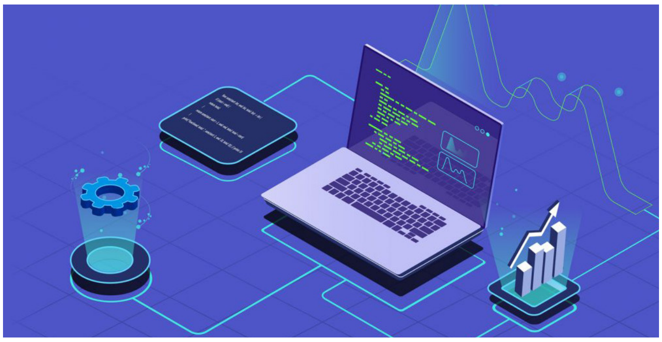
I/O Streams – 4 hrs

- Introduction
- Hierarchy of streams
- IO Stream, Byte Stream, Character Streams
- BufferedInputStream, BufferedOutputStream
- Reader and Writer class
- BufferedReader, PrintWriter
- Serialization

Collection Framework – 6 hrs

- basics, hierarchy
- legacy classes – Vector, Queue, Stack, Enumeration, Dictionary, Properties
- List, ArrayList, LinkedList
- Set, HashSet, TreeSet,
- Map, HashMap, TreeMap,
- Generics
- Annotations
- Boxing / Unboxing
- Enums

Introduction to functional style of programming – 1 hrs



Mini application – discussion / implementation – 10 hrs

Tools

- Javadoc
- javap
- jar

IDE Tools

- eclipse
- myeclipse

Highlights of Training

- Industry experienced Professional
- Hands-on experience with Project orientation
- Interview based Questions

Our Approach

- Proper Communication with the student
- Course Ends With One Live Project
- Test on Each Topics
- Certification Overview
- Interview and resume preparation
- Discussion on Real Time Scenarios

ADVANCED JAVA

Database and SQL Fundamentals

Relational Databases and SQL
Database, Schema, Tables, Columns and Rows
DDL & DML
Sequences , Stored Procedures
Using SQL Terminals

Introduction

J2EE Overview
Why J2EE?
J2EE Architecture
J2EE APIs
J2EE Containers

Servlet

HTML Forms
Servlets Overview
Servlet Lifecycle: init(), service(), destroy()
GenericServlet, ServletRequest, and ServletResponse
HttpServletRequest, HttpServletResponse and HttpServlet
Request-response, headers, GET, POST

JDBC

JDBC Fundamentals
Advanced JDBC
Introduction to Row Sets
JDBC classes and the java.sql
Connecting to databases
JDBC design patterns

Session Management

HTTP as a stateless protocol
Hidden form fields
Cookies
session tracking
Http Session
Exception handling and error pages
Directives
Single Thread Model interface

JavaServer Pages

Overview
JSP architecture , JSP tags and JSP expressions
Fixed Template Data , Lifecycle of a JSP
Model View Controller (MVC)
Model 1/Model 2 Architecture

Data Sharing among servlets & JSP
Request, application, session and page scope
JSP implicit objects

RMI (Remote Method Invocation)

RMI overview
RMI architecture
Example demonstrating RMI

Java Beans

Enterprise Bean overview
Types of enterprise beans
Advantages of enterprise beans
The Life Cycles of Enterprise Beans
Working with Session Beans
Statefull vs. Stateless Session Beans
Working with Entity Beans
Message Driven Beans

JNDI

JNDI overview
JNDI API
Context operations
Using JNDI in J2EE applications

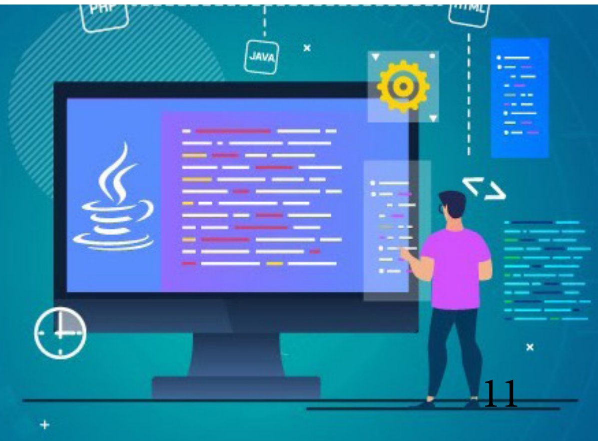
Struts Framework

What is Struts?
Struts Architecture
Struts classes – ActionForward, ActionForm,
ActionServlet, Action classes
Understanding struts-config.
Understanding Action Mappings
Struts flow with an example application

Struts Tiles Framework
Struts Validation Framework
Internationalizing Struts Application
Struts with Message Resources

Other J2EE Technologies

Hibernate
Spring framework
JSF
Web services
SOA, SOAP, AXIS, WSDL, UDDI,
Endpoint & Callback
AJAX



Advanced Framework with Microservices

Spring core – 4 hrs

- Types of containers
- Xml configuration
- Core annotations
- Component – scan
- What can be injected into a bean
- Understanding core annotations

Spring MVC – 6 hrs

- Handler Mapping
- Controllers
- View resolvers
- Validators
- interceptors

Spring DAO – 4 hrs

- Jdbc templates
- Exception hierarchy
- Spring ORM – 2 hrs
- Hibernate template
- Integration with hibernate

Spring AOP – 2 hrs

- Point cut
- Advisors
- Types of advices

Spring – webservices (Restful API) – 3 hrs

- Spring security – 4 hrs
- Spring JPA – 2hrs

Hibernate Training Syllabus

Introduction – 4 hrs

- Introduction to ORM (Object Relational Management)
- Benefits of ORM
- Contrast JDBC with ORM
- Hibernate Jar files
- Hibernate Architecture
- SessionFactory , Session, Transaction, Query

- Understanding config files .hibernate.xml, .hbm.xml

Performing CRUD operations

- load, get, save, saveorupdate, delete

HQL – 2 hrs

- working with various SQL commands
- accessing more than one object from the table

Mapping – 6 hrs

- Inheritance
- Collection
- Composition
- Association

Cache – 3 hrs

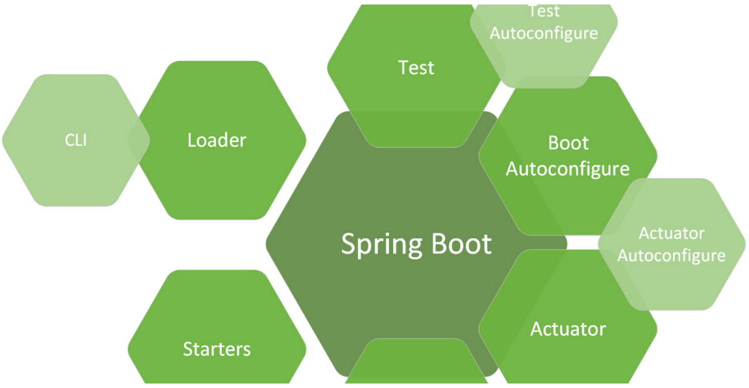
- Level – I and II
- Using third party to configure L2 cache

Criteria – 3 hrs

- Interfaces / classes
- implementation
- Note : an application using the above technologies –
- Domains : Telecom , LMS, etc...

Spring Boot Training Syllabus

Approx. 15 – 20 hrs



- Introduction / benefits
- Spring Boot overview
- Bootstrapping an Boot application –
- Initializer
- Auto Configuration
- Spring Boot annotations
- Spring Boot properties
- Spring Boot profiles
- Accessing data with Spring Boot and h2 database
- Configuring Spring Boot with spring MVC application
- Building a Restful web application with Spring Boot

Highlights of Training

An application based on Spring boot – Either migration / new one

Industry experienced Professional
Hands-on experience with Project orientation
Interview based Questions

Advanced Java Training Syllabus

Fundamentals – 2 hrs

- Client, Server, Port, Application, Web Server, Application Server, Web Client and Web Application
- Installation of Java and Understanding PATH and

CLASSPATH

- Installation of Eclipse
- Installation of Tomcat and structure of Tomcat
- Linking any Web Server with Eclipse
- Static Resource Access from Tomcat

Servlets Syllabus – 10 hrs

Servlet Internals and Servlet Programming

- What are Servlets and why do we need Servlets?
- Servlet Development with Eclipse

– Developing Servlets
– Compiling Servlets
– Deployment and Invoking of Servlets

- Purpose of Servlet Mapping and web.xml
- Servlet Architecture

– Detailed Architecture of Servlets
– Discussion of GenericServlet, HttpServlet Interfaces
– Programming examples

- Servlet Life Cycle
- HTTP Internals
- Detailed discussion of all parameters of request and response packets
- Discussion of HttpServletRequest and HttpServletResponse interfaces

Session Tracking API
– Practical visualization of Http Session objects and Session table
– Session Tracking API
– Session Tracking Programming

- Cookies, URL Rewriting and Hidden Form Fields
- ServletConfig Object
- ServletContext Object
- RequestDispatcher Object
- Filters

– Why do we need filters?
– Filter, FilterChain, FilterConfig interfaces
– Programming examples

- Application / Attribute Lifecycle Listeners
- Why do we need Lifecycle Listeners?
- ServletContext and ServletContext Attribute Listeners
- Session and Session Attribute Listeners
- Programming examples

JSP Syllabus – 10 hrs

Java Server Pages(JSP) Internals and Programming

- What is JSP? Why do we need JSP?
- JSP development with Eclipse
- JSP Life Cycle
- Predefined Objects of JSP
- request, response, out and session objects
- config, application, pagecontext objects
- Scripting Tags (Writing Java code inside JSP)
- Directives
- @page, @include directives
- Action tags of JSP
- Difference between jsp:forward, jsp:include and @include tags
- Programming examples to illustrate each tag
- Expression Language
- Why do we need EL?
- Reading scoped attributes using EL
- Reading bean properties using EL
- Reading collections using EL
- Operators of EL
- Programming examples to illustrate each usage
- JSTL (JSP standard tag library)
- Why do we need JSTL?
- Looping tags
- Control tags
- Database tags
- 118N tags
- Programming examples to illustrate each category of tags
- Custom tags
- Custom tag creation based on Servlets
- Custom tag creation based on JSP
- concept of tld file
- programming examples to illustrate each concept

Java Beans – 2 hrs

- What are Java beans and why do we need Java beans?
- Action tags for java beans
- Application of Java beans

MVC Architecture 2 hrs

- Why do we need MVC?
- Discussion of pros and cons of Servlets, JSP and JavaBeans
- Programming example to illustrate Business

logic, Controller logic, Presentation logic and Bean creation

JDBC – 6 hrs

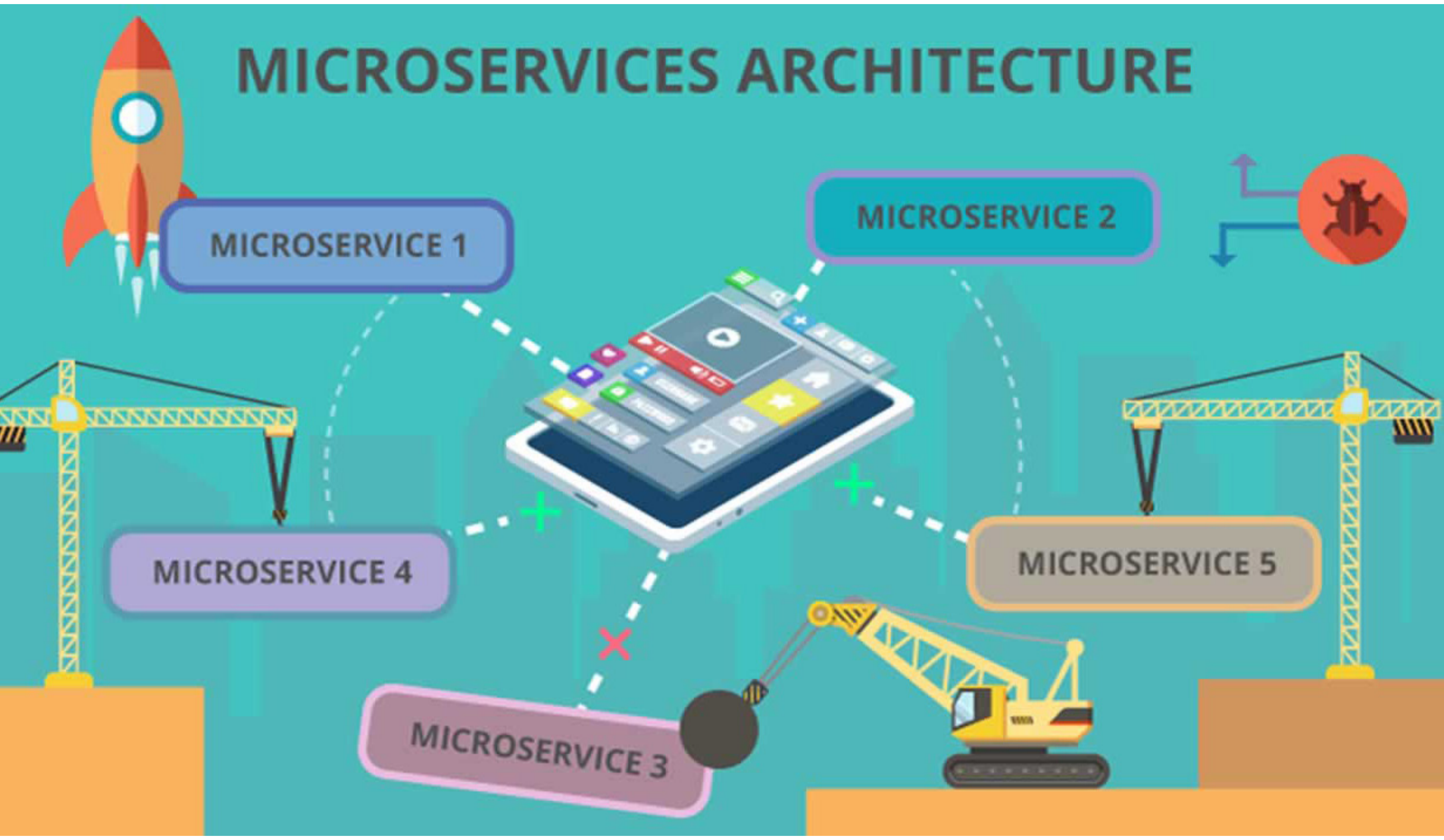
Java Data Base Connectivity (JDBC) internals and programming

- What is JDBC ? Why do we need JDBC ?
- Discussion of JDBC driver types and Driver Manager
- Connection object
- Statement types and Discussion of each type of Statement Objects
- ResultSet Types and Discussion of each type
- Stored Procedure Access from java applications
- Transaction Programming with JDBC
- Savepoint and RowSet
- DataSource and Connection Pooling
- Batch Updates
- JNDI

Explanation on Usage of important Practical Tools

- Eclipse
- Jboss, Weblogic, Tomcat

Note : An application based on Traditional MVC using the above technologies



MEAN STACK

Node.js

Introduction

- Node.js Background
- Getting Node.js
- Demo: Installing Node on Linux/Windows with NVM
- Node's Event Loop
- Node Conventions for Writing Asychronous Code
- Modules, require() and NPM
- Introduction, Accessing Built-in Modules
- Finding 3rd Party Modules via NPM

Express

- Introduction
- Setting up Express
- Running Express
- Debugging Options
- Static Files – Public Directory
- Introduction, Making Web Requests in Node
- Building a Web Server in Node
- Demo: Building a Web Server in Node

Routing

- Introduction
- Navigation
- Routing
- Rendering
- Separate Files
- Router Functions

Authentication

- Introduction to Authentication
- Auth Routes
- Passport
- Local Strategy

Events and Streams

- Introduction
- Events and the EventEmitter class
- Readable and Writable Streams, the Pipe function
- Demo: Readable and Writable Streams
- Demo: Piping Between Streams

Accessing the Local System

- The Process Object
- Demo: The Process object

- Interacting with the File System
- Demo: Interacting with the File System
- What is a Buffer?
- The OS Module

Socket.IO

- Realtime Interaction with Socket.IO
- Demo: Socket.IO

Testing and Debugging

- Introduction, The Assert Module
- Demo: The Assert Module
- Testing with Mocha and Should.js
- Demo: Mocha and Should.js

Scaling Your Node Application

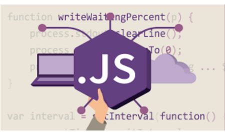
- Introduction, The Child Process Module
- Demo: The “exec” function
- Demo: The “spawn” function
- Demo: The “fork” function
- Scaling with Node’s Cluster Module
- Demo: Building a Clustered Web Server

MongoDB

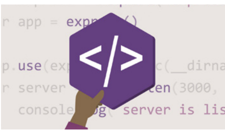
Introduction

- Downloading
- Package Content
- First Run
- Command Line Options
- Verify Server
- Getting Help in the Shell

Saving Data



- Introduction
- Overview
- Storage
- BSON
- Saving Documents
- Collections
- Document Id
- ObjectId
- Insert
- Insert with Id
- Complex Document
- Save Danger
- Update Command
- Update Demo
- Set Operator
- Unset Operator
- Rename Operator
- Push Operator
- Pull Operator
- Pop Operator
- Array Type
- Multi Update
- Find And Modify
- Query With Sort



Finding Documents

- Introduction
- Overview
- find()
- Equality
- Projection
- Comparison
- $not
- $in
- Arrays
- $all
- $nin
- Dot Notation
- Sub-Document
- null and $exists
- And
- Cursor
- sort()
- limit()
- skip()
- findOne()

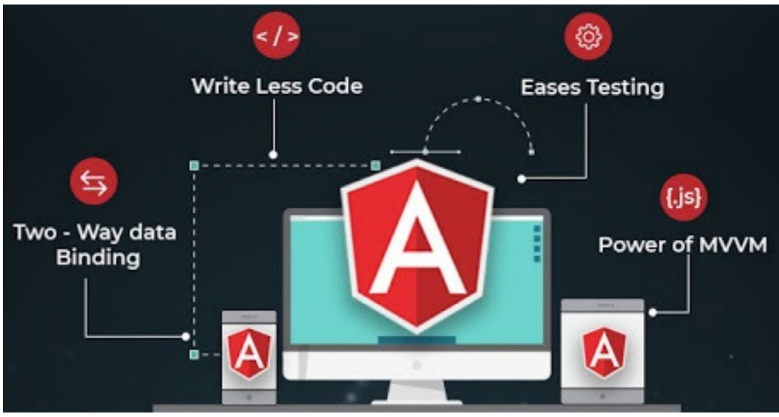
Deleting Document

- remove
- deleteOne
- deleteMany

Indexing

- Introduction
- Overview
- Scan Is Bad
- Index Theory
- Sort Uses Index
- Index Types
- Create Index
- system.indexes collection
- explain()
- dropIndex()
- Index Name

Angular.js



Getting Started with Angular

- Introduction
- Introduction to TypeScript
- Comparing Angular to AngularJS
- A Conceptual Overview of Angular
- Getting Started with the Angular CLI
- Bootstrapping an Angular App
- A Brief Look at the App Module
- Accessing Static Files

Creating and Communicating Between Angular Components

- Introduction
- Creating Your First Data-bound Component
- Using External Templates
- Communicating with Child Components Using @Input
- Communicating with Parent Components Using

@Output
- Using Template Variables to Interact with Child Components
- Styling Components
- Exploring Angular’s CSS Encapsulation
- Adding a Site Header

Exploring the Angular Template Syntax

- Introduction
- Interpolation, Property Bindings, and Expressions
- Event Bindings and Statements
- Repeating Data with ngFor
- Handling Null Values with the Safe-Navigation Operator
- Hiding and Showing Content with ngIf
- Hiding Content with the [Hidden] Binding
- Hiding and Showing Content with ngSwitch
- Styling Components with ngClass
- Styling Components with ngStyle

Creating Reusable Angular Services

- Introduction
- Why We Need Services and Dependency Injection
- Creating Your First Service
- Wrapping Third Party Services

Routing and Navigating Pages

- Introduction
- Adding Multiple Pages to Your App
- Adding Your First Route
- Accessing Route Parameters
- Linking to Routes
- Navigating from Code
- Guarding Against Route Activation
- Guarding Against Route De-activation
- Pre-loading Data for Components
- Styling Active Links

Collecting Data with Angular Forms and Validation

- Introduction
- Using Models for Type Safety
- Creating Your First Template-based Form
- Using the Data from Your Template-based Form

Creating Your First Reactive Form

- Validating Reactive Forms
- Using Multiple Validators in Reactive Forms
- Diving Deeper into Template-based Forms
- Editing Data with Two-way Bindings

- Diving Deeper into Reactive Forms
- Creating Custom Validators

Communicating Between Components

- Introduction
- Passing Data into a Child Component
- Passing Data out of a Child Component

Reusing Components with Content

- Projection
- Introduction
- Content Projection
- Multiple Slot Content Projection

Displaying Data with Pipes

- Introduction
- Using Built-in Pipes
- Creating a Custom Pipe
- Sorting and Filtering Overview
- Creating a Filtering Display
- Filtering Data
- Sorting Data
- Summary

Understanding Angular's Dependency

- Injection
- Introduction
- Using Third Party Global Services
- Using Angular's InjectionToken
- Using Angular's @Inject Decorator
- The useClass Provider
- The useExisting and useFactory Providers

Creating Directives and Advanced

- Components in Angular
- Introduction
- Implementing the Session Search
- Adding jQuery
- Creating a Modal Component
- Creating Directives
- Binding an ID
- Routing to the Same Component
- Using the @ViewChild Decorator
- Creating Settings on Components

Communicating with the Server Using HTTP, Observables

- Introduction
- Preparing to Store Data on

the Server
- Moving Data Storage to the Server
- Listening to Resolved Data Changes
- Using POST and PUT
- Using QueryString Parameters
- Using DELETE
- Integrating Authentication with the Server
- Persisting Authentication Status Across Page Refreshes
- Saving User Data to the Server
- Implementing Logout

Unit Testing Your Angular Code

- Introduction
- Installing Karma6m 16s
- Unit Testing Services8m 16s
- Testing Mock Calls5m 44s
- Testing Components with Isolated Tests

Testing Angular Components with

- Integrated Tests
- Introduction
- Setting up for Integrated Tests
- Testing Components with Deep Integrated Tests
- Creating Mock Services
- Using DebugElement
- Testing Components with Shallow Integrated Tests

Taking an Angular App to Production

- Introduction
- Linting Overview
- Installing TSLint in VSCode
- Using TSLint with VSCode
- Going to Production – Overview
- Creating your First Build
- Basic Deployment
- Build Flags
- The Effects of Prod Mode
- Optimistic Bundle Downloading

Which companies use mean stack?

- 53 companies reportedly use MEAN in their tech stacks, including Accenture, Fiverr, and UNIQLO.
- Accenture.
- Fiverr.
- UNIQLO.
- Sisense.
- Raindrop.io.
- AngularClass.
- Tajawal.
- Vungle.
- More..

HTML | CCS5 | JAVA SCRIPT

PART 1 – HTML 5

Module 1: Web Programming

- Introduction
- Architecture of a website
- Different technologies in making the website
- Web Development Introduction

Module 2: HTML-Introduction

- History of HTML
- What you need to do to get going and make your first HTML page
- What are HTML Tags and Attributes?
- HTML Tag vs. Element
- HTML Attributes

Module 3: HTML-Basic

- Formatting Tags
- HTML Basic Tags
- HTML Formatting Tags
- HTML Color Coding

Module 4: HTML-Grouping

- Using Div Span
- Div and Span Tags for Grouping

Module 5: HTML-Lists

- Unordered Lists
- Ordered Lists
- Definition list

Module 6: HTML-Images

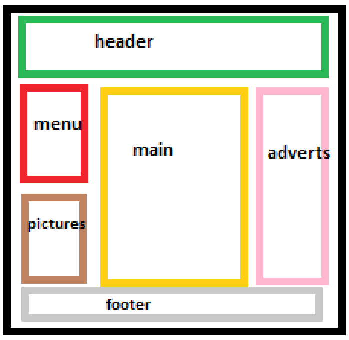
- Image and Image Mapping

Module 7: HTML-Hyperlink

- URL – Uniform Resource Locator
- URL Encoding

Module 8: HTML-Table

- < table >
- < th >
- < tr >
- < td >
- < caption >
- < thead >
- < tbody >
- < tfoot >
- < colgroup >
- < col >



Module 9: HTML-Iframe

- Using Iframe as the Target

Module 10: HTML-Form

- < input >
- < textarea >
- < button >
- < select >
- < label >

Module 11: HTML-Headers

- Title
- Base
- Link
- Styles
- Script
- Meta

Module 12: HTML-

Miscellaneous
HTML Meta Tag
XHTML
HTML Deprecated Tags & Attributes

PART 2 – CSS 3

```
body {
  color: #000;
  font: 16px Times, serif;
}
h1 {
  color: #f09;
  font-size: 24px;
}
```

Module 1: CSS3-Introduction

- Benefits of CSS
- CSS Versions History
- CSS Syntax
- External Style Sheet using < link >
- Multiple Style Sheets
- Value Lengths and Percentages

Module 2: CSS3-Syntax

- CSS Syntax
- single Style Sheets
- Multiple Style Sheets
- Value Lengths and Percentages

Module 3: CSS3-Selectors

- ID Selectors
- Class Selectors
- Grouping Selectors
- Universal Selector
- Descendant / Child Selectors
- Attribute Selectors
- CSS – Pseudo Classes

Module 4: CSS3-Color

- Background Cursor
- background-image
- background-repeat
- background-position
- CSS Cursor

Module 5: CSS3-Text Fonts

- color
- background-color
- text-decoration
- text-align
- vertical-align
- text-indent

- text-transform
- white space
- letter-spacing
- word-spacing
- line-height
- font-family
- font-size
- font-style
- font-variant
- font-weight

Module 6: CSS3-Lists Tables

- list-style-type
- list-style-position
- list-style-image
- list-style
- CSS Tables
- border
- width & height
- text-align
- vertical-align
- padding
- color

Module 7: CSS3-Box Model

- Borders & Outline
- Margin & Padding
- Height and width
- CSS Dimensions

Module 8: CSS3-Display

- Positioning
- CSS Visibility
- CSS Display
- CSS Scrollbars
- CSS Positioning
- Static Positioning
- Fixed Positioning
- Relative Positioning
- Absolute Positioning
- CSS Layers with Z-Index

CSS Floats

The float Property
The clear Property
The clear fix Hack

JAVASCRIPT

```
<script type="text/javascript">
        switch (new Date().getDay()) {
          case 6:
            text = "Friday";
            break;
          case 0:
            text = "Sunday";
            break;
          default:
            text = "Choose Your Day";
        }
</script>
```

Section 1: Introduction, Setup and Syntax

- Introduction
- Setup
- What is Javascript
- Course Exercise Files
- Practice App & Display Functions

Assignments 1:

- Using a Code Editor
- Variables
- Arrays

Assignments 2:

- Practice App

Section 2: Control Structures, Functions, Math & Events

- If Statements
- Loops
- Math
- Functions
- Even Handlers Part:1 (Example:
- Event Handlers Part:2

Assignments:03

- Practice App – If Statements and Loops
- Practice App – Math & Functions
- Practice App – Events

Section 3: Advance Scenario #01

- Printing Arrays
- Text Output Function with a Loop
- Built-in Functions (Displaying the Time In Javascript)
- Math Built-in Functions

Section 4: Advance Scenario #02

- Assigning, Adding and Removing Elements from Arrays
- Inserting and Extracting From Arrays more functions
- ForloopArray Checking with Challenge Part #1
- ForloopArray Checking with Solution Part #2
- Setting Up Flags and Using Booleans
- Where to go next?
- Quick Syntax Reference/ Resource File

AWS CLOUD DEVELOPMENT

IAM

- IAM
- IAM Lab

EC2

- EC2 101
- EC2 Lab
- How To Use Putty (Windows Users Only)
- Elastic Load Balancer
- Route 53 Lab
- CLI Demo Lab
- EC2 with S3 Role Lab
- How to Encrypt an EBS Volume Attached to EC2 Lab
- RDS 101
- RDS Lab
- RDS Multi-AZ and Read Replicas
- Elasticache 101
- EC2 Summary

S3

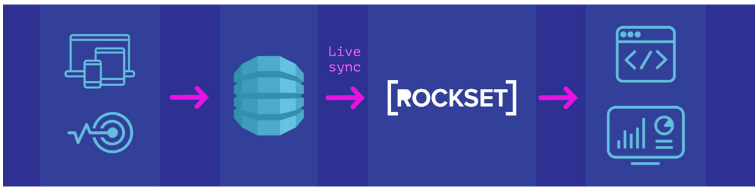
- S3
- S3 Security
- S3 Policies
- S3 Encryption
- Set Up Encryption On an S3 Bucket
- CORS Configuration Lab
- CloudFront
- CloudFront Lab
- S3 Performance Optimization
- S3 Performance Update
- S3 Summary

Serverless Computing

- Serverless
- Lambda
- Build a Simple Serverless Website with Route 53, API Gateway, Lambda and S3
- Version Control with Lambda
- Using Polly to Help Your Exam Lab – Part 1 &amp; 2
- Make an Alexa Skill Lab
- Step Functions
- Advanced API Gateway
- Serverless Summary

DynamoDB

- Introduction to DynamoDB
- Creating a DynamoDB Table Lab
- Indexes Deepdive
- Scan vs Query API Call
- DynamoDB Provisioned Throughput
- DynamoDB Accelerator (DAX) ElastiCache



KMS and Encryption on AWS

- KMS
- KMS API Calls
- KMS Envelope Encryption
- KMS Exam Tips

Other AWS Services

- SQS
- Simple Notification Service
- SES vs SNS
- Kinesis
- Kinesis Lab
- ElasticBeanstalk
- Deploying Apps With
-
- ElasticBeanstalk
- Updating ElasticBeanstalk
- Advanced ElasticBeanstalk
- RDS & ElasticBeanstalk
- Other AWS Services Summary
- Other AWS Services Quiz

Developer Model

- What is CI/CD?
- CodeCommit
- CodeCommit Lab
- CodeDeploy
- CodeDeploy Lab 1, 2 & 3
- CodePipeline
- CodePipeline Lab
- Advanced CodeDeploy the AppSpec File
- Docker & CodeBuild Lab 1 & 2

Advanced IAM

- Web Identity Federation
- Cognito User Pools
- Inline Policies vs Managed Policies vs Custom Policies

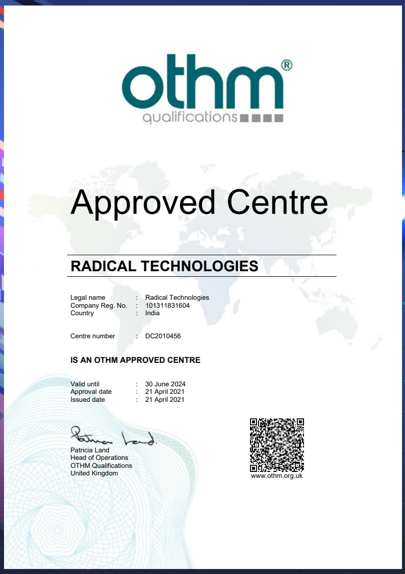
PUNE
BANGALORE
KOCHI

8055223360
8103400400
8882400500

www.radicaltechnologies.co.in

Training Queries:
training@radicaltechnologies.co.in

Radical Technologies
Project Based IT Training & Certification Centre



othm qualifications

Approved Centre

RADICAL TECHNOLOGIES

Legal name	:	Radical Technologies
Company Reg. No.	:	101311831604
Country	:	India
Centre number	:	DC2010456

IS AN OTHM APPROVED CENTRE

Valid until	:	30 June 2024
Approval date	:	21 April 2021
Issued date	:	21 April 2021

Patricia Land
Head of Operations
OTHM Qualifications
United Kingdom

www.othm.org.uk

Aundh | Hinjewadi | Pimple Saudagar | Kharadi | Karve Road | Magarpatta | Banglore | Kochi